Detecting BGP Anomalies Using Machine Learning Techniques

Qingye Ding, Zhida Li, Prerna Batta, and Ljiljana Trajković

Communication Networks Laboratory, Simon Fraser University, Vancouver, British Columbia, Canada

BGP Anomalies

- Border Gateway Protocol (BGP) plays an essential role in routing data between Autonomous Systems (ASes).
- BGP anomalies affect Internet servers and hosts and are manifested by anomalous traffic behavior.
- They may be detected by analyzing collected traffic data and by generating various classification models.
- Machine learning techniques are the most common approaches for classifying BGP anomalies.
- We employ two supervised machine learning algorithms:
  - Support Vector Machine (SVM)
  - Long Short-Term Memory (LSTM).

Feature Selection

- Datasets are collected from the Route Views project, the Réseaux IP Européens (RIPE) Network Coordination Centre (NCC), and from BCNET.
- We extracted 37 features from BGP update messages originated from AS 513.
- Three cases of well-known anomalies are considered: Slammer, Nimda, and Code Red I.
- 10 features were selected using minimum Redundancy Maximum Relevance (mRMR) algorithms:
  - Mutual Information Deference (MID), Mutual Information Quotient (MIQ), and Mutual Information Base (MIBASE).

Support Vector Machine

- Support Vector Machine (SVM) is a supervised learning model for classification and regression tasks.
- SVM algorithm learns a classification hyperplane (decision boundary) by maximizing the minimum distance between data points belonging to various classes.
- We use soft-margin SVMs that allow certain data points to be misclassified.
- The hyperplane is acquired by solving a loss function with constraints:

$$C \times \Sigma_{n=1}^{N} \zeta_n + \frac{1}{2}||w||$$

$$t_n y(x_n) \geqslant 1 - \zeta_n, n = 1, ..., N$$

- C > 0 controls the trade-off between the margin and the penalty term $\frac{1}{2}\|w\|$.
- t_n denotes the target value.
- $y(x_n)$ is the training model.
- ζ_n is the slack variable.

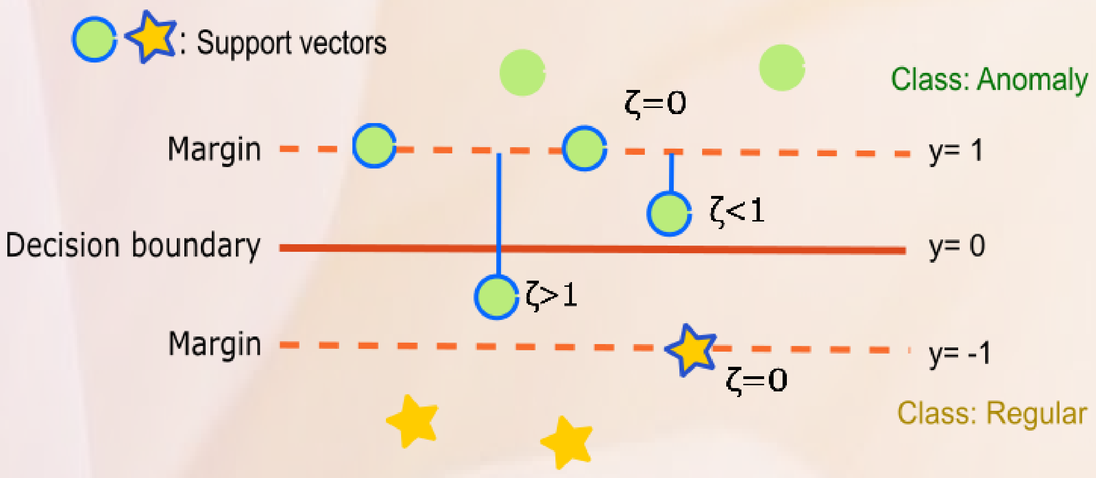
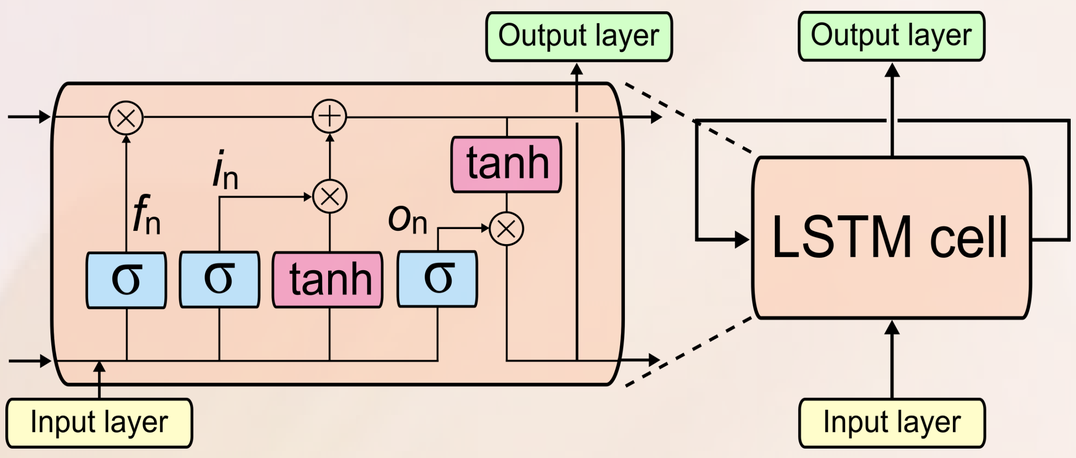


Illustration of the soft margin SVM. Data points with circles are support vectors. Shown are correctly (ζ≤1) and incorrectly (ζ>1) classified data points.

Long Short-Term Memory (LSTM) Neural Network

- The LSTM approach employs a special form of the Recurrent Neural Networks.
- LSTM is capable of connecting time intervals to form a continuous memory.
- LSTM cell, also called the “memory block”:
  - forget gate f_n: discards the useless memories according to the cell state
  - input gate i_n: controls the information that will be updated in the LSTM cell
  - output gate o_n: controls the output.



Repeating modules for the LSTM neural network.
Shown are the input layer, LSTM cell, and output layer.

Experimental Procedure

- The SVM and LSTM models are generated using both unbalanced and balanced datasets.
- Unbalanced data: number of regular data is larger than number of anomalies.
- Balanced data: contain all anomalies and randomly selected equivalent number of regular data.
- Step 1: Train and test the SVM and LSTM models using 37 features.
- Step 2: Select 10 most relevant features. Train and test SVM models. Skip this Step for LSTM models.
- Step 3: Evaluate performance of SVM and LSTM models.
- Step 4: Tune SVM and LSTM model parameters to achieve the best performance.

Classification Environment

- Machine learning tools:
  - SVM: SVM^{light}, a library developed in C language.
    - Tune the cost factor and the trade-off parameter that controls the training error and the margin.
  - LSTM: PyBrain, a modular Machine Learning library for Python.
    - Use to generate LSTM models with 37-dimensional inputs, 1 hidden layer, and 1-dimensional outputs.

Performance Evaluation

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \qquad \text{F-score} = 2 \times \frac{\text{precision} \times \text{sensitivity}}{\text{precision} + \text{sensitivity}}$$

$$\text{precision} = \frac{TP}{TP + FP} \qquad \text{sensitivity} = \frac{TP}{TP + FN}$$

- The SVM and LSTM training and testing datasets:

Model	Training dataset	Testing dataset
SVM_1 and LSTM_1	Slammer and Nimda	Code Red I
SVM_2 and LSTM_2	Slammer and Code Red I	Nimda
SVM_3 and LSTM_3	Nimda and Code Red I	Slammer

- Accuracy and F-score using the SVM_2 models for unbalanced and balanced datasets:

Unbalanced Datasets				
	Accuracy			F-score
	Testing Dataset	RIPE	BCNET	Testing Dataset
SVM_u 2 37 Features	67.46 %	52.85 %	46.39 %	68.60 %
SVM_u 2 MID	70.79 %	58.40 %	50.69 %	69.54 %
SVM_u 2 MIQ	65.33 %	58.54 %	51.81 %	64.88 %
SVM_u 2 MIBASE	66.74 %	53.40 %	48.33 %	69.97 %

Balanced Datasets				
	Accuracy			F-score
	Testing Dataset	RIPE	BCNET	Testing Dataset
SVM_b 2 37 Features	69.26 %	51.81 %	44.86 %	72.32 %
SVM_b 2 MID	60.96 %	55.35 %	54.31 %	63.36 %
SVM_b 2 MIQ	51.89 %	32.43 %	43.68 %	62.63 %
SVM_b 2 MIBASE	67.10 %	65.14 %	55.00 %	63.84 %

- The accuracy and F-score using LSTM models for unbalanced and balanced datasets:

Unbalanced Datasets				
	Accuracy			F-score
	Testing Dataset	RIPE	BCNET	Testing Dataset
$LSTM_u$ 1	89.58 %	65.49 %	57.30 %	23.33 %
$LSTM_u$ 2	60.00 %	51.53 %	50.80 %	54.87 %
$LSTM_u$ 3	63.15 %	56.47 %	58.55 %	24.68 %

Balanced Datasets				
	Accuracy			F-score
	Testing Dataset	RIPE	BCNET	Testing Dataset
$LSTM_b$ 1	45.04 %	60.48 %	62.78 %	16.72 %
$LSTM_b$ 2	63.16 %	44.27 %	53.58 %	58.16 %
$LSTM_b$ 3	61.24 %	55.00 %	48.20 %	27.48 %

- Performance improvements by using balanced datasets:
  - SVM_2 models achieve the best F-score (72.32 %) with 37 features.
  - LSTM models have the best performance (58.16 %).
- Improved F-score results from careful extraction of features and the use of balanced training datasets.
- Poor performance of LSTM_1 and LSTM_3 models may be caused by noisy data as well as small number of employed LSTM cells.

Conclusion

- Using balanced datasets to train the SVM models leads to better F-scores than the results previously reported that used unbalanced datasets.
- In case of the unbalanced datasets, the accuracy is higher due to the large number of the regular testing data.
- Using the SVM classifier may be a feasible approach for detecting BGP anomalies in communication networks.

References

- N. Al-Rousan and Lj. Trajković, “Machine learning models for classification of BGP anomalies,” in *Proc. IEEE Conf. on High Performance Switching and Routing*, *HSPR* 2012, Belgrade, Serbia, June 2012, pp. 103–108.
- Y. Li, H. J. Xing, Q. Hua, X.-Z. Wang, P. Batta, S. Haeri, and Lj. Trajković, “Classification of BGP anomalies using decision trees and fuzzy rough sets,” in *Proc. IEEE Trans. Syst., Man, Cybern.,* San Diego, CA, October 2014, pp. 1312-1317.
- RIPE NCC. [Online]. Available: http://www.ripe.net .
- University of Oregon Route Views project. [Online]. Available: http:// www.routeviews.org/.
- SVM^{light}: Support Vector Machines. [Online]. Available: http://svmlight.joachims.org .
- S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, October 1997.
- PyBrain: The Python Machine Learning Library. [Online]. Available: http://pybrain.org/.

SFU SIMON FRASER UNIVERSITY THINKING OF THE WORLD

SMC 2016
IEEE International Conference on Systems, Man, and Cybernetics
October 2016, Budapest, Hungary

Communication Networks Laboratory
http://www.ensc.sfu.ca/~ljilja/cnl